Gridftp & Globus Online on ANI Testbed

By Dave Dykstra, with some initial testing done by Raman Verma and guidance by Gabriele Garzoglio, all of FNAL
Last updated 06 October 2011

In the interests of brevity and understandability, this report does not go into detail on the processes used to find the optimal conditions for best performance, or on all the testing attempts used to find them. Suffice it to say that many variations were tried and this report just refers to the variations that supplied the best results on the given hardware and network configuration. The intention for this report is to include enough detail for someone else to be able to reproduce the results on the same hardware.

1. Test hardware and network

The Advanced Networking Initiative testbed has been installed to prepare for testing 100Gbit/s cross-country networking. So far the most bandwidth available is 30 Gbit/s to a computer at BNL (called here “bnl-1”). The bandwidth is all on 10Gbit/s interfaces. Bnl-1 has four usable 10Gbit/s interfaces currently, two 10Gbit/s non-interfering connections to a co-located computer called here “bnl-2” (round-trip ping time 0.1ms) and two 10Gbit/s connections to a computer elsewhere in the Long Island metropolitan area called here “newy-1” (round-trip ping time 2ms). The four interfaces weren’t completely independent, however, and only three at a time could be used in order to get the best throughput.

All the gridftp transfers mentioned in this report were initiated on bnl-1 and data was sent from the other computers to bnl-1. All 3 computers have respectably fast disk arrays that were benchmarked at 14Gbit/s write speed, but they were still not adequate to keep up with 30Gbit/s so transfers were all done to /dev/null. The files transferred were small enough to be cached in the operating system’s filesystem RAM buffer cache, so disk read speed wasn’t a factor.

The testbed hardware is available only via a Virtual Private Network; it is not directly accessible via the internet and therefore Globus Online. This was accommodated by using a Virtual Machine at Fermilab (called here “gateway”) that ran the VPN software and accepted gridftp control port connections from the internet which it forwarded to the three testbed computers using xinetd forwarding. The three testbed computers further forwarded those connections with xinetd forwarding to their own gridftp control ports (port 2811) bound on the various 10Gbit/s interfaces. The gridftp servers were then able to identify the endpoints for directly transferring the data at as high a speed as possible. The xinetd options used for forwarding were:

```
flags = REUSE
socket_type = stream
wait = no
user = root
redirect = <ip> <port>
log_on_failure += USERID
```

```
   per_source = UNLIMITED
   cps = 2048 1
   instances = UNLIMITED
```

The gridftp servers themselves were run from xinetd and had the following default options as set in the Open Science Grid distribution of gridftp:

```
    instances   = UNLIMITED
    cps          = 400 10
    per_source  = 300
```

2. Data set

The idea for the data set was to use 100GB of transfers split into three parts: big files, medium files, and small files. The total data set consisted of about 16GB of files on disk, and they were sent repeatedly to make up the desired amount of data for each size of file. There were 21 files ranging in size by power of 2 between 8KB (2^13 bytes) and 8GB (2^33 bytes). The big file data set sent 30GB in 3 files ranging by power of 2 from 2GB (2^31 bytes) to 8GB (2^33 bytes) for a total of 14GB of disk space repeated in 7 file transfers. The medium file data set sent 40GB in 9 files ranging by power of 2 from 8MB (2^23 bytes) to 1GB (2^30 bytes) for a total of 2040MB of disk space repeated in 180 file transfers. The small file data set sent 30GB in 10 files ranging in powers of 2 from 8KB (2^13 bytes) to 4MB (2^23 bytes) for a total of 4088KB of space repeated in 42240 file transfers. When the entire 100GB was sent it was a combination of the 3 data sets so there were a total of 42432 file transfers.

3. The tests

All the tests used three 10Gbit/s interfaces, so they sent each data set separately over each of the three paths, tripling the total amount of data sent. Using each of three different gridftp control methods (described below), measurements were done for each of the four different data sets: all files (100GB*3=300GB), big files (30GB*3=90GB), medium files (40GB*3=120GB), and small files (30GB*3=90GB). There was no difficulty in sending the big data set using two 10Gbit/s interfaces between bnl-1 and newy-1 and one 10Gbit/s interface between bnl-1 and bnl-2, but for an unexplained reason the small file data set was consistently much slower over those interfaces than when using two 10Gbit/s interfaces locally between bnl-1 & bnl-2 and one 10Gbit/s interface between bnl-1 and newy-1. The medium set was inconsistently significantly slower using the two interfaces between bnl-1 & newy-1, so the tests for both small and medium files always used the two interfaces between bnl-1 & bnl-2 and one between bnl-1 & newy-1, but the big files were always sent with the two interfaces between bnl-1 & newy-1 and one local between bnl-1 & bnl-2. The tests with all files were sent the same ways as the three separate pieces, with the exception that Globus Online required all files in a single transfer to go over a single interface so the Globus Online all files test always used two 10 Gbit/s interfaces between bnl-1 & bnl-2 and one between bnl-1 and newy-1.

The all files tests were staggered over the three different interfaces so that one interface had big+medium+small, the second had medium+small+big, and the third had small+big+medium.

The three types of tests were:

1. Local: three parallel globus-url-copy commands initiated on bnl-1. The performance tuning parameters in each case were:
   big: -fast -cc 2 -p 4
   med: -fast -cc 2
   small: -fast -cc 64, and add -pp for bnl-1 to newy-1 transfers (-pp saved a few seconds)
2. FNAL: three parallel globus-url-copy comands initiated on another Virtual Machine at Fermilab, connecting through the gateway machine. The performance tuning parameters were:
   big: -fast -cc 4 -p 4
   med: -fast -cc 5 -pp
   small: -fast -cc 64 -pp (-pp saved a lot of time in this case)
3. Globus Online: three transfer jobs initiated using gsissh to cli.globus.org as fast as they could right after each other. Timing was taken from the email reports, starting from the start of the first job to the end of the last one. The performance tuning parameters were:
   big & med: auto-tune
   small & all: --perf-cc 16 --perf-pp 32 (the maximum allowed)

4. The Results

Tests were done three times each and the total number of seconds for each test is the first line of each entry in the table. After that comes a calculation of thoughput in gigabytes per second based on the average of the three measurements.

	all, 100GB*3	big, 30GB*3	med, 40GB*3	small, 30GB*3
Local	120s,119s,116s 300GB/118.3s =2.5GB/s	33s,30s,31s 90GB/31.3s =2.9GB/s	39s,37s,37s 120GB/37.7 =3.2GB/s	53s,52s,53s 90GB/52.7s =1.7GB/s
FNAL	203s,192s,197s 300GB/197.3s =1.5GB/s	34s,34s,34s 90GB/34s =2.6GB/s	39s,39s,39s 120GB/39s =3.1GB/s	167s,164s,170s 90GB/167s =0.54GB/s
Globus Online	380s,381s,389s 300GB/383.3s =0.78GB/s	58s,59s,66s 90GB/61s =1.5GB/s	40s,40s,49s 120GB/43s =2.8GB/s	275s,287s,307s 90GB/289.7s =0.31GB/s

Observations: Small files have a reasonable overhead for locally-intiated transfers, but unreasonably high over the wide area control channels. Globus Online auto-tuning appears to do better on the medium files than the large ones.

5. Followup Globus Online test, increased limits on perf parameters

The Globus Online organization supplied a new test version of their system (their “QA” system) that alloweded increased concurrency and pipelining parameters, up to 100 each. This was tested a few weeks after the first tests. First the previous Globus Online tests were re-run for comparison, then the new version was tested. The only test setup difference was that the second 10Gbit/s connection between bnl-1 and newy-1 was not available, so the big file test used the same paths as the others (that is, one 10Gbit/s connection between bnl-1 and newy-1 and two between bnl-1 and bnl-2). Since the Globus Online small files test was considerably slower this time, the small files test controlled from Fermilab was re-done as well. No "--perf" parameters were used for the measurements on the small files test with the new test Globus Online system, because all parameters attempted only slowed the results down.

	all, 100GB*3	big, 30GB*3	med, 40GB*3	small, 30GB*3
Globus Online re-tested	405s,391s,409s 300GB/401.7s =0.75GB/s	56s,56s,57s 90GB/56.3s =1.6GB/s	46s,41s,42s 120GB/43s =2.8GB/s	489s,468s,481s 90GB/479.3s =0.19GB/s
FNAL re-tested partially				170s,166s,163s 90GB/166.3s =0.54GB/s
Globus Online new test system	411s,407s,445s 300GB/421s =0.71GB/s	57s,57s,57s 90GB/57s =1.6GB/s	41s,47s,41s 120GB/43s =2.8GB/s	388s,348s,348s 90GB/361.3s =0.25GB/s

Observations: the repeated Globus Online tests for the "small" dataset mysteriously took 65% longer this time. It seems that the Globus Online service was either changed or it was under a heavier load. The other 3 types of tests were about the same as before, as was the FNAL-initiated test of small files.

On the new Globus Online test system, increasing either --perf-cc and --perf-pp on the small files test only seemed to slow down further the higher the values of parameters that were used. There wasn't enough time to do complete complete measurements; this is based on comparison results of bits per second printed out from the cli "wait" command.